行列を使ったグレブナー基底計算 (Computer Algebra : Design of Algorithms, Implementations and Applications)

AUTHOR(S):

鈴木, 晃

CITATION:

鈴木, 晃. 行列を使ったグレブナー基底計算 (Computer Algebra : Design of Algorithms, Implementations and Applications). 数理解析研究所講究録 2009, 1666: 9-13

ISSUE DATE:

2009-10

URL:

http://hdl.handle.net/2433/141084

数理解析研究所講究録
第1666巻 2009年 9-13

行列を使ったグレブナー基底計算

鈴木 晃
AKIRA SUZUKI*
神戸大学
KOBE UNIVERSITY†

Abstract

線形代数に於けるガウスの消去法とブッフバーガーアルゴリズムの類似性については, これまでも論じられており, またグレブナー基底の計算のために線形代数を用いる手法についても, F_4, F_5 を含め, これまでいくつか論じられてきた. 本報告では項順序の制約を外す事により, ガウスの消去法を全面的に利用しつつも, 効率的にグレブナー基底に類した代数構造を計算するための手法を論じる.

1 はじめに

グレブナー基底を計算するブッフバーガーアルゴリズムと, 線形代数に於けるガウスの消去法の類似性については Lazard [3] 等により研究されてきた. ブッフバーガーアルゴリズムそのものは, 本質的には, S-多項式生成と単項簡約の組み合わせであるが, この内, 特に単項簡約の部分にガウスの消去法を用いたものが Faugère の F_4 や F_5 [1, 2] である.

本報告では, S-多項式生成も特殊な単項簡約と見なす事で, ブッフバーガーアルゴリズム全体を行列の計算で行う手法について考察する. これにより, 単項簡約を持たない数式処理システムに於いても, グレブナー基底の計算が可能となる. グレブナー基底を線形代数の枠組みの中で計算する方法については, 近年再び報告者を含む複数の研究者により研究が開始されており, それら方法の効率化への貢献が期待される.

以下, K を体, $\bar{X} = \{X_1, \ldots, X_n\}$ を変数, $T(\bar{X})$ を $\bar{X}$ による項とする. また, 特にことわりない限り, a, b, c を K の元, f, g, h を $K[\bar{X}]$ の多項式, α, β, γ を $T(\bar{X})$ の項とする.

2 mdeg順序とそのグレブナー基底基底

線形代数に於いてグレブナー基底を計算するにあたり, 多項式をベクトルで表現するのが自然であろう. 先ず, 適切な有限 $T \subseteq T(\bar{X})$ とその上の全順序 $<$ を考え, $T = \{t_1, \ldots, t_m\}$ $(t_1 > t_2 > \cdots > t_m)$ とおく. そこに表れる項が全て T に含まれるような多項式 $f \in K[\bar{X}]$ を $f = a_1 t_1 + \cdots a_m t_m$ $(a_1, \ldots, a_m \in K)$ と表現し, $\mathrm{vec}(f) \in K^m$ を $\mathrm{vec}(f) = (a_1, \ldots, a_m)$ で定義し, 多項式の計算を線形空間 $\langle T \rangle_K$ もしくは K^m の中で行う.

一般的な項順序は以下のように定義される.

定義 1 (項順序) 項全体の集合 $T(\bar{X})$ 上の全順序 $<$ が**項順序**であるとは以下を満たす時に言う:

*本研究は科研費 (20500013) の助成を受けたものである.
†sakira@kobe-u.ac.jp

1. $<$ は整列順序であり,

2. $\alpha < \beta$ であれば任意の $\gamma \in T(\bar{X})$ に対して $\alpha\gamma < \beta\gamma$ である.

多項式からベクトルへの変換に使われた順序 $<$ を項順序として指定すると, 与えられた有限多項式集合 $F \subseteq \langle T \rangle_K$ に対して $\{\mathrm{vec}(f) : f \in F\}$ から生成される行列を考えると, そのガウス消去の結果により F の元達の互いの単項簡約の一部がそこで計算される事がわかる. ここで $|T| = m$ を小さくする事がガウス消去の計算を軽くするために有効であると考えられるが, そのためには, T が, ある $\alpha_0 \in T(\bar{X})$ に対して $T = \{\beta \in T(\bar{X}) : \beta \leq \alpha_0\}$ と表現されている事が望ましい. 一方で, ブッフバーガーアルゴリズムでは単項簡約に加えて S-多項式が重要な役割を持つが, $\{\beta \in T(\bar{X}) : \beta \leq \alpha_0\}$ という形式の T の内で S-多項式の計算は制限される. この困難を克服するために以下のように $T(\bar{X})$ 上の順序を定義する.

定義 2 (mdeg) 項 $\alpha = X_1^{a_1} X_2^{a_2} \cdots X_n^{a_n} \in T(\bar{X})$ に対して $\mathrm{mdeg}(\alpha) = \max\{a_1, a_2, \ldots, a_n\}$ と定義する. また, 多項式 $f \in K[\bar{X}]$ に対しても $\mathrm{mdeg}(f)$ を $\mathrm{mdeg}(f) = \max(\mathrm{mdeg}[\mathrm{terms}(f)])$ で定義する. 但し $\mathrm{terms}(f) \subseteq T(\bar{X})$ は f にあらわれる項全体の集合とする.

定義 3 (mdeg 順序) 項全体の集合 $T(\bar{X})$ 上の全順序 $\prec$ が mdeg **順序**であるとは以下を満たす時に言う:

1. $\prec$ は整列順序であり,

2. $\beta \neq 1$ であれば $\alpha \prec \alpha\beta$ であり,

3. $\mathrm{mdeg}(\alpha) < \mathrm{mdeg}(\beta)$ であれば $\alpha \prec \beta$ である.

以下のように mdeg 順序は項順序の条件を満たさない点には注意すべきである.

例 4 $T(X,Y)$ 上の *mdeg* 順序 $\prec$ を固定する. $f = X^2 + Y \in \mathbb{Q}[X,Y]$ に対して $\mathrm{mdeg}(f) = \mathrm{mdeg}(X^2) = 2 > 1 = \mathrm{mdeg}(Y)$ であり, f の $\prec$ に関しての先頭項は $\mathrm{ht}_\prec(f) = X^2$ である. 一方で, $Y^2 f = X^2Y^2 + Y^3$ を考えると $\mathrm{mdeg}(Y^2 f) = \mathrm{mdeg}(Y^3) = 3 > 2 = \mathrm{mdeg}(X^2Y^2)$ であり, $Y^2 f$ の $\prec$ に関しての先頭項は $\mathrm{ht}_\prec(Y^2 f) = Y^3$ となり, $\mathrm{ht}_\prec(Y^2 f) \neq Y^2 \mathrm{ht}_\prec(f)$ となる.

なお, 実装の際には以下のような形で mdeg 順序を定義するのが自然であろう.

例 5 (coherent mdeg 順序) 順序 $<$ を $T(\bar{X})$ 上の項順序とする. この時, $T(\bar{X})$ 上の *mdeg* 順序 $\prec$ を以下のように定義できる:

1. $\mathrm{mdeg}(\alpha) < \mathrm{mdeg}(\beta)$ であれば $\alpha \prec \beta$,

2. $\mathrm{mdeg}(\alpha) = \mathrm{mdeg}(\beta)$ かつ $\alpha < \beta$ であれば $\alpha \prec \beta$.

この時 $\prec$ が $<$ に対して一意に決まる事と, $\prec$ が *mdeg* 順序の条件を満たす事は容易に確認でき, $\prec$ を $<$ に coherent な *mdeg* 順序であると呼ぶ.

なお, mdeg 順序は項順序の条件を満たさないが, mdeg 順序に対してもグレブナー基底は定義され, また, 変更されたブッフバーガーアルゴリズムで可能で計算可能である. 先ず単項簡約を定義する.

定義 6 (単項簡約) $\prec$ を $T(\bar{X})$ 上の *mdeg* 順序とする. 多項式 $f + a\alpha \in K[\bar{X}]$ を $\alpha \notin \mathrm{terms}(f)$ かつ $a \in K \setminus \{0\}$ なるものとする. 更に $g \in K[\bar{X}]$, $b \in K$, $\beta \in T(\bar{X})$ を $b\alpha = \mathrm{ht}_\prec(\beta g)$ なるものとする. この時 $a\alpha + f$ *reduces to* $f' = a\alpha + f - ab^{-1}\beta g$ *modulo* βg *w.r.t.* $\prec$ と言い, $a\alpha + f \longrightarrow_{\beta g} f'$ と記す.

ここで項順序に関する単項簡約のように「modulo g」としない点に注意が必要である. 実際, mdeg 順序では modulo αg から modulo g は導かれない. また, $f \longrightarrow_g f'$ から $\alpha f \longrightarrow_{\alpha g} \alpha f'$ も導かれない.

この時, mdeg 順序に関するグレブナー基底は以下のように定義できる.

定義 7 (グレブナー基底) $F, G \subseteq K[\bar{X}]$ とし, $\prec$ が $T(\bar{X})$ 上の整列順序であるとする. この時, G が F の ($\prec$ に関する) **グレブナー基底**であるとは, $F \subseteq \langle G \rangle_{K[\bar{X}]}$ であり, 任意の $f \in F$ に対して $g \in G$ と $\alpha \in T(\bar{X})$ が $\mathrm{ht}_\prec(f) = \mathrm{ht}_\prec(\alpha g)$ と取れる時に言う.

なお $\prec$ が項順序で $F = \langle G \rangle_{K[\bar{X}]}$ である時にはこの定義は通常のグレブナー基底のものと同等である.

与えられた有限 $F \subseteq K[\bar{X}]$ と mdeg 順序 $\prec$ に対して, 若干変更したブッフバーガーアルゴリズムにより, グレブナー基底を計算できるが, 例 4 にあるように, mdeg 順序では多項式の内で先頭項になるものは掛ける項によって異なるため, S-多項式の生成にも注意する必要がある.

定義 8 (S-多項式) $f, g \in K[\bar{X}]$ で, $a\alpha$ を f の, $b\beta$ を g の単項とする. この時, $\gamma = \mathrm{lcm}(\alpha, \beta)$ として, $\mathrm{SPol}(f, g; \alpha, \beta) = b(\gamma/\alpha)f - a(\gamma/\beta)g$ とする.

このような定義を用いて, $f \in K[\bar{X}]$ に対して, 何らかの項 $\alpha \in T(\bar{X})$ で $\mathrm{ht}_\prec(\alpha f) = \alpha\beta$ となる, つまり先頭項となる可能性のある全ての $\beta \in \mathrm{terms}(f)$ を考慮して, S-多項式を生成すべきである.

3 mdeg 順序グレブナー基底の行列による計算

前節で述べたように, 変形ブッフバーガーアルゴリズムにて mdeg 順序に関するグレブナー基底は計算可能であるが, 本節では, mdeg 順序を導入した理由である行列計算によるグレブナー基底計算のより具体的な方法について述べる.

項の有限集合 $T \subseteq T(\bar{X})$ に対し, $T = \{t_1, t_2, \ldots, t_l\}$ $(t_1 \succ t_2 \succ \cdots \succ t_l)$ と記し, 多項式 $f = a_1 t_1 + \cdots + a_l t_l \in \langle T \rangle_K$ に対して, $\mathrm{vec}_\prec(f) = (a_1, \cdots, a_l) \in K^l$ と同型写像 $\mathrm{vec}_\prec : \langle T \rangle_K \to K^{|T|}$ を定義する.

例 9 $X > Y$ である *grlex* 項順序に *coherent* な *mdeg* 順序 $\prec$ を考え, $T = \{\alpha \in T(\bar{X}) : \mathrm{mdeg}(\alpha) \leq 2\}$ とする. ここで T に含まれる項を $\succ$ の順に列挙すると $X^2Y^2 \succ X^2Y \succ XY^2 \succ X^2 \succ Y^2 \succ XY \succ Y \succ X \succ 1$ であるので, $\mathrm{vec}_\prec(2X^2Y - XY^2 + 5X) = (2, 0, -1, 0, 0, 0, 0, 5, 0) \in \mathbb{Q}^9$ である.

次に $T \subseteq T(\bar{X})$ の形を mdeg により限定する. 自然数 d に対して $\mathrm{TB}(d) = \{\alpha \in T(\bar{X}) : \mathrm{mdeg}(\alpha) \leq d\}$ と置く.

定義 10 (mult) 多項式 $f \in K[\bar{X}]$ と自然数 d (もしくは $d = \infty$) に対して $\mathrm{mult}(f, d) = \{\alpha f : \alpha \in T(\bar{X}), \mathrm{mdeg}(\alpha f) \leq d\}$ とし, 更に $F \subseteq K[\bar{X}]$ に対して $\mathrm{mult}(F, d) = \bigcup_{f \in F} \mathrm{mult}(f, d)$ と定義する.

この時, d が自然数であれば $\mathrm{mult}(f, d)$ は有限集合であり, 更に F が有限集合であれば $\mathrm{mult}(F, d)$ も有限であり, また計算可能である事もわかる.

ブッフバーガーアルゴリズムで使う計算の本質的な部分は, 何度か述べたように, S-多項式生成と単項簡約であるが, それらを更に分解して考えると

- 多項式 f への単項 $a\alpha$ の乗算 $((f, a\alpha) \mapsto a\alpha f)$ と,
- 多項式 f, g 同士の加算 $((f, g) \mapsto f + g)$

にて構成されている. 一方で, 上記の mult と行列のガウス消去では

mult: 多項式 f への項 α の乗算 $(f \mapsto \{\alpha_1 f, \alpha_2 f, \ldots\})$ が,

ガウス消去: 多項式 f,g と係数 $a,b \in K$ の乗算と加算 $((f,g,a,b) \mapsto af+bg)$ が,

計算され, 本質的には mult とガウス消去の組合せにて (何らかの $\langle \mathrm{TB}(d) \rangle_K$ 内に於ける) ブッフバーガーアルゴリズムが内包される. 従って mult とガウス消去を互いに繰り返すという以下のアルゴリズムが行列によるグレブナー基底計算の基本となる. 但し, `LinearReduce` は入力された多項式の集合を行列と同一視した (行基本変形による) ガウス消去を行うアルゴリズムであるとする.

アルゴリズム `CloseAndLReduce`

Input: F : $K[\bar{X}]$ の有限部分集合, d : $F \subseteq \langle \mathrm{TB}(d) \rangle_K$ なる自然数,
$<$: $T(\bar{X})$ 上の mdeg 順序もしくは項順序
Output: R : $\langle F \rangle_{K[\bar{X}]} = \langle R \rangle_{K[\bar{X}]}$ なる有限集合

```
R := F;
n := |R|;
n' := 0;
while n' ≠ n do
  n' := n;
  R := LinearReduce(mult(R,d), <);
  n := |R|;
end while
return R;
end.
```

このアルゴリズムの停止性は容易に証明できる. また与えられた有限 $F \subseteq K[\bar{X}]$ と自然数 d に対して R を `CloseAndLReduce(`F`, <)` の出力であるとすると, R =`LinearReduce(`R`, <)` かつ $R = \mathrm{mult}(R,d)$ である事もわかる. 出力された R がグレブナー基底をなしているかどうかの判定には次を用いる.

定義 11 (width) 各項 $\alpha = X_1^{a_1} \cdots X_n^{a_n} \in T(X_1,\ldots,X_n)$ に対して, $\mathrm{pow}(\alpha,i) = a_i$ とする. 更に, 各 $f \in K[\bar{X} = \{X_1,\ldots,X_n\}]$ と $i = 1,\ldots,n$ に対して, $\mathrm{width}(f,i) = \max\{\mathrm{pow}(\alpha,i) - \mathrm{pow}(\beta,i) \ : \ \alpha,\beta \in \mathrm{terms}(f)\}$ と定義する.

この時, 以下がわかる. 証明はこれを省略する.

補題 12 $\prec$ が $T(\bar{X})$ 上の *mdeg* 順序もしくは項順序とする. $G \subseteq K[\bar{X}]$ と自然数 d が R =`LinearReduce(`G`,` $\prec$`)` かつ $R = \mathrm{mult}(G,d)$ を満たすとする. 更に G_0 及び自然数 c が $G_0 \subseteq G \cap \langle \mathrm{TB}(c) \rangle_K$ であり G_0 が G のグレブナー基底であり, 各 $g \in G_0$ と $i = 1,\ldots,n$ に対して $c + \mathrm{width}(g,i) \leq b$ であると仮定する. この時 G_0 は $\prec$ に関してイデアル $\langle G_0 \rangle_{K[\bar{X}]}$ に対するグレブナー基底である.

これを用いると次のアルゴリズムで, 与えられた $F \subseteq K[\bar{X}]$ から $\langle F \rangle_{K[\bar{X}]}$ のグレブナー基底を計算できる事がわかる. 詳細は [4] を参照されたい.

アルゴリズム `LAGroebnerBasis`

Input: F : $K[\bar{X}]$ の有限集合, $\prec$: $T(\bar{X})$ 上の mdeg 順序又は項順序
Output: G : $\prec$ に関する $\langle F \rangle_{K[\bar{X}]}$ のグレブナー基底

G := F;
$G \subseteq \mathrm{TB}(c_0)$ となるよう c_0 を設定;
b := 0;
c := c_0;
d := 0;
G_0 := $\emptyset$;
while $c+d > b$ **do**
    b := $c+d$;
    G := **CloseAndLReduce**(G, b, $\prec$);
    G_0 := $\{g \in G : (\forall g' \in G \setminus \{g\})\, \mathrm{ht}_\prec(g') \not| \mathrm{ht}_\prec(g)\}$;
    G_1 := $G \cap \langle \mathrm{TB}(c_0) \rangle_K$;
    $G_0 \cup G_1 \subseteq \mathrm{TB}(c)$ となるよう c を設定;
    d := $\max\{\mathrm{width}(g,i) : g \in G_0 \cup G_1,\ i = 1,\ldots,n\}$;
end while
return G_0;
end.

なお, このアルゴリズムで G_1 は, **while** ループ中で $G_0 \cup G_1$ がイデアル $\langle F \rangle_{K[\vec{X}]}$ を生成し, 更には, アルゴリズムの最後で G_0 がそのイデアルを生成する事を, 保証するために用いられている.

4 最後に

本報告では, 線形代数の範囲内でグレブナー基底に類するものを効率的に計算するために, mdeg 順序とそれに関するグレブナー基底を定義し, それを計算するアルゴリズムを提示した. 但し, 各変数にウェイトを導入する事でより効率的な計算が可能となる筈であるが, 本報告ではそれを省略した.

一般に, mdeg 順序に関するグレブナー基底は項順序に関するグレブナー基底とはならず, 逆もまた成り立たない. しかし, mdeg 順序に関するグレブナー基底も, イデアルメンバーシップ問題など項順序に関するグレブナー基底の応用の多くには利用できる. 一方で, あくまでも項順序に関するグレブナー基底を求めたいという要請も, 当然, 多いであろう. mdeg 順序に関するグレブナー基底を項順序に関するグレブナー基底に変換する手法については, 項順序と mdeg 順序のハイブリッドも含め, いくつかの案があるが, いずれも問題点を残したままで, 有力な候補を見つけられていない. これらの議論についても, 本報告では省略した.

今後は, 項順序に関するグレブナー基底への変換手法の確立に加え, mdeg 順序に関するグレブナー基底の実装, 計算実験などが必要であろう.

参 考 文 献

[1] Faugère, J.C. (1999) *A new efficient algorithm for computing Gröbner bases (F_4)*. J. Pure and Applied Algebra, **139**(1–3), 61–88.

[2] Faugère, J.C. (2002) *A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)*. Proc. ISSAC '02, 75–83.

[3] Lazard, D. (1983) *Gröbner-Bases, Gaussian elimination and resolution of systems of algebraic equations*. EUROCAL 1983, 146-156.

[4] Suzuki. A. (∞) *Computing Gröbner bases within Linear Algebra*. submitting.